

<https://helda.helsinki.fi>

---

## Full Charge Capacity and Charging Diagnosis of Smartphone Batteries

Hoque, Mohammad Ashraful

2017-11-01

---

Hoque , M A , Siekkinen , M , Koo , J & Tarkoma , S 2017 , ' Full Charge Capacity and Charging Diagnosis of Smartphone Batteries ' , IEEE Transactions on Mobile Computing , vol. 16 , no. 11 , pp. 3042-3055 . <https://doi.org/10.1109/TMC.2017.2688321>

---

<http://hdl.handle.net/10138/327020>

<https://doi.org/10.1109/TMC.2017.2688321>

---

acceptedVersion

---

*Downloaded from Helda, University of Helsinki institutional repository.*

*This is an electronic reprint of the original article.*

*This reprint may differ from the original in pagination and typographic detail.*

*Please cite the original version.*

# Full Charge Capacity and Charging Diagnosis of Smartphone Batteries

Mohammad A. Hoque, Matti Siekkinen, Jonghoe Koo, and Sasu Tarkoma

**Abstract**—Full charge capacity (FCC) refers to the amount of charge a battery can hold. It is the fundamental property of smartphone batteries that diminishes as the battery ages and is charged/discharged. We investigate the behavior of smartphone batteries while charging and demonstrate that battery voltage and charging rate information can together characterize the FCC of a battery. We propose a new method for accurately estimating FCC without exposing low-level system details or introducing new hardware or system modules. We further propose and implement a collaborative FCC estimation technique that builds on crowd-sourced battery data. The method finds the reference voltage curve and charging rate of a particular smartphone model from the data and then compares with those of an individual device. After analyzing a large data set towards a crowd-sourced rate vs. FCC model, we report that 55% of all devices and at least one device in 330 out of 357 unique device models lost some of their FCC. For some old device models, the median capacity loss exceeded 20%. The models further enable debugging the performance of smartphone charging. We implement an algorithm, called BatterySense, which utilizes crowd-sourced rate to detect abnormal charging performance, estimate FCC of the device battery, and detect battery changes.

**Keywords**—Battery, Full Charge Capacity, State of Charge, Fuel Gauge, Charging rate, Voltage.

## I. INTRODUCTION

Smartphone users frequently encounter battery and energy problems. From the popular Internet blogs [1], [2], [3], we have identified two issues that are increasingly being reported by the users; sudden drop in the battery level and disgraceful shutdown of the device even with high battery levels being reported to the user (even at 80%) while discharging. These observations are reported across different smartphone models, and even for laptops. This disgraceful shutdown may bar users from their scheduled phone activities and result in data loss. From a user's perspective, the remaining battery life of a smartphone may even converge to the monetary value [4]. Many smartphone manufacturers have recently introduced battery replacement programs that cover batteries that have a reduced capacity, typically below 80% [5], [6]. The current smartphone battery discussion pertains to the following questions: *Why does the battery level fluctuate? Is the battery*

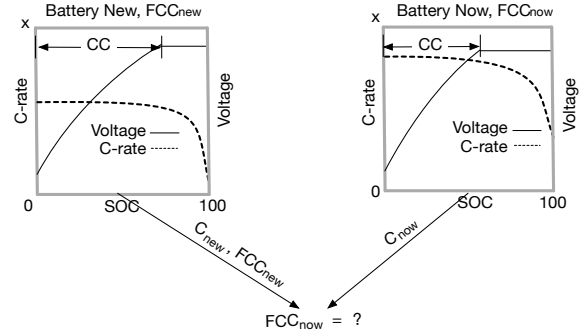


Fig. 1. The comparison of voltage curves and C-rates during the constant current (CC) phase of charging towards finding the recent FCC of a smartphone battery.

*faulty? Is the problem due to an operating system upgrade or installing/upgrading an application?*

The answers to the earlier questions are related to the smartphone full charge capacity (FCC), as we demonstrated in prior work [7]. FCC is the maximum amount of charge an empty battery can hold. As the battery of a smartphone ages, the full charge capacity decreases with the utilization. FCC is typically modeled as a function of the number of charging cycles [39] or measured with Coulomb counting technique [14]. This functionality or measurement capability resides inside a smart battery. The battery shares this estimate as percentage with the hosting device, such as smartphone (see Section II). Therefore, an indication of the capacity loss along with the battery level would allow more accurate state of charge (SOC) estimation. For instance, a cheaper and simple voltage-based fuel gauge can estimate FCC while charging and apply it for estimating SOC instead of applying complex charging cycle-based learning. Similarly a simple Coulomb counter-based fuel gauge can calibrate FCC even when the battery is partially charged. The other applications are power consumption modeling, sophisticated energy-aware scheduling mechanisms by the system and different applications.

In this article, we examine the performance of smartphone batteries and present a novel FCC estimation technique that can infer the FCC and FCC loss. The approach works whether a smart battery is capable to measure FCC or not, and therefore enables any device to estimate the FCC of any attached battery. We discover that the battery voltage curve and battery capacity relative charging rate, i.e., C-rate, during a distinct phase of charging can characterize the FCC of a smartphone battery given the reference curves of the new battery. Based on these findings, we devise a new FCC estimation method as shown in Figure 1. Our evaluation suggests that the estimation error is

Mohammad A. Hoque and Sasu Tarkoma are with the Department of Computer Science, University of Helsinki, Finland. Email: first-name.lastname@cs.helsinki.fi

Matti Siekkinen is with the Department of Computer Science and Engineering, Aalto University, Finland. Email: matti.siekkinen@aalto.fi

Jonghoe Koo is with the Department of Electrical and Computer Engineering and INMC, Seoul National University, Seoul, Korea. Email: jhkoo@mwnl.snu.ac.kr

limited to 10% of the true value. This is an application-layer approach and does not involve any new hardware. It brings new information, such as FCC, to OS designers or users for resource efficient scheduling. The fuel gauge manufacturers can design cheaper and reliable fuel gauges without additional hardware cost and complexity. It is also possible to diagnose the performance of other energy source, i.e., the charger, of smartphones.

For facilitating large-scale battery health analytics, we also present a crowd-sourced approach that works with battery voltage and charging rate information solely obtained from a crowd-sourced data set. We derive a reference voltage curve and a C-rate from the data set for each model using a statistical approach and then we apply the FCC estimation method that compares the charging rate of a device with the model-specific reference rate towards finding the capacity loss of the devices that contribute to such a data set. We demonstrate that this method works relatively well for most models found in the Carat data set [8]. Furthermore, we discovered that 40% of all the 9560 devices had some capacity loss, and that within 357 unique models at least one device in 333 models suffered from capacity loss. We summarize our contributions as follows.

- We investigate the behavior of battery voltage and charging rate while charging as the FCC of a smartphone battery decreases. Consequently, we propose a new battery capacity estimation model. The model is validated with AC and USB2.0 constant current chargers and the model yields estimates with an accuracy of 90% or more according to our evaluation. We further evaluated the performance of the model with partial charging, ambient temperature while charging, and different SOC estimation techniques.
- We introduce a crowd-sourced FCC estimation technique and demonstrate how such a method can be validated. Although the accuracy of the approach depends on the diversity of the community, a study with the large scale Carat data set shows that our collaborative technique estimates the reference rates of popular smartphone models within a 10% error margin. We also examine the presence of devices with reduced FCC in the Carat data set.
- Finally, in order to realize the practical impact of the FCC estimation models, we propose and implement an algorithm for Android devices. The performance of smartphone batteries also depends on the charging performance of the device and user behavior in charging. Since C-rate is the ratio of the charging current and FCC, it is possible to characterize the charger is being used for charging. Finally, along with reporting FCC to the user, we combine the above two approaches as an algorithm for Android devices, called BatterySense. BatterySense finds the FCC loss, and charging issues, such as abnormal fast and slow charging.

The paper is organized as follows: Section II presents a smartphone power management primer. In Section III, we investigate battery voltage and charging current behavior while charging smartphones. Section IV presents the model and it is validated in Section V. In Section VI, we present the

crowd-sourced battery analytic data set and the FCC estimation technique. The functionality of BatterySense is discussed in Section VII. Section VIII presents the related work. The future research directions are outlined in Section IX before concluding the paper.

## II. BACKGROUND

A battery, a fuel gauge chip, and a charging controller are the fundamental components of a smartphone's power management system. In this section, we first describe how these components function and then highlight the scope of this article.

*Charging Controller:* The charging controller is responsible for charging the battery. Among various techniques [25], Constant Current-Constant Voltage (CC-CV) charging is the widely utilized approach. During CC period, the charging current remains constant until the battery voltage reaches a specified maximum (4.2/4.35V). Over the CV period, the charging current is trickled until the battery is fully charged. The charging terminates when the charging rate reduces to 0.07C or to a lower cut-off charging current specified by the manufacturer [10]. In this case, C is the rate that is relative to the battery capacity as follows: If the capacity of a battery is 2600 mAh and it takes one hour to fully charge/discharge a battery, it means that 2600 mA rate is equivalent to 1 C for that battery. Similarly, 0.5 C-rate is equivalent to 1300 mA for that battery. C-rate is the widely accepted metric to represent the charging/discharging rate and extensively utilized in determining battery properties, such as SOC and FCC modeling.

*Battery Pack:* Smartphones are powered with single cell battery packs. Along with the cell, a battery pack may also host SOC/FCC measurement functionalities and such batteries are called smart batteries. The battery pack may also include a protection mechanism to guard against higher voltage and current from the device. The FCC of a battery decreases as it ages and through progressive chemical reactions. Graphite is the common material used as anode in Lithium-Ion batteries and there are multiple anode-cathode pairs in a battery. As the battery is being charged, the oxidation of the graphite constructs a layer, called passive surface layer [11]. If the outer shell leaks, then the oxidation happens faster due to moisture and the capacity loss accelerates.

*Fuel Gauge:* The chip with SOC estimation functionality is often called fuel/gas gauge, which may be distributed between the battery pack and the host system, i.e., smartphone [9]. The smartphone only queries the battery for the supported information, such as SOC and battery voltage. The SOC is the runtime estimate of the battery charge. A SOC value of 0 and 100 imply an empty and fully charged battery, respectively. The most common approach to estimate SOC is to use open circuit voltage (OCV) with a number of look-up tables. A voltage-based fuel gauge may also combine both OCV and load voltage to estimate SOC or energy drain [12], [13]. The second approach is Coulomb counting, which introduces a sense resistor on the charge and discharge path. Finally,

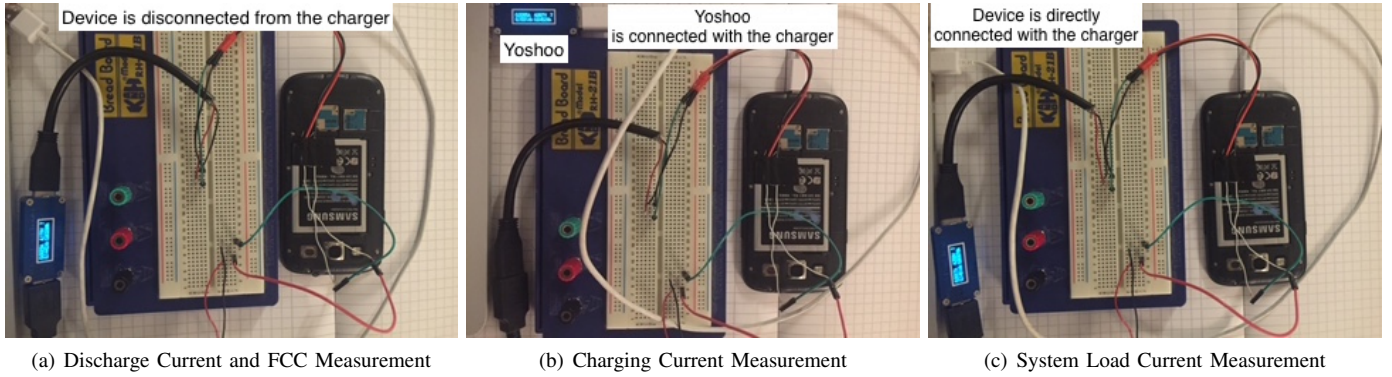


Fig. 2. **Smartphone battery discharging current and FCC measurements, charging current, and system load current measurements with Yoshoo power monitor.** Yoshoo is a simple USB3.0 pass-through power measurement tool. In the first setting, Yoshoo is connected with the charger and measures the charging current. In the second and third settings, the Yoshoo is connected with a USB3.0 male and female terminal. The female terminal is attached with the battery and the male terminal is attached with the device pins. Only, the +/- terminals of the battery are insulated from the corresponding device pins. The remaining terminals are in contact with the device pins so that the device can sense the presence of the battery.

modern fuel gauges may also combine voltage measurements with Coulomb-counter [29].

A fuel gauge chip may be able to measure or estimate the FCC along with SOC. However, voltage-based fuel gauges cannot estimate or measure the FCC. Therefore, a such fuel gauge chip may use the number of charging cycles that the battery has gone through to estimate the FCC. On the other hand, Coulomb counter-based fuel gauges use sense resistors to measure FCC. However, the measurements can be used internally by the fuel gauge to recalibrate the SOC and may not be shared with the smartphone.

This article proposes and validates a new software-based FCC estimation technique. Unlike the above approaches, *our approach neither requires complex learning nor additional hardware. The method equally performs in devices with both voltage and Coulomb counting-based fuel gauges. In addition, we apply the same method to detect the performance of the connected chargers.*

### III. SMARTPHONE BATTERY CAPACITY & CHARGING BEHAVIOR

In this section, we measure the FCC and investigate the performance of smartphone batteries while *charging*. We used Samsung Galaxy S4 (GT-I9505) and S3 (GT-I9300) devices with twelve batteries, as their batteries are replaceable. The FCC of a new Galaxy S4 and S3 batteries are 2600 and 2100 mAh, respectively. Both devices are shipped with fuel gauges that can estimate SOC either with voltage-based method that uses both OCV and load voltage or a approach combined with Coulomb-counter. However, we verified that both devices use the former approach<sup>1</sup>. The aim of the experiments is to realize

<sup>1</sup>These devices are equipped with MAX17047/17050 fuel gauge chip that implements load voltage and OCV-based SOC estimation technique, called ModelGauge, and also has a coulomb counter. The fuel gauge can be configured during the kernel compile time to use either ModelGauge or a combined method with the coulomb counter. The mobile system exposes the measured current value from the fuel gauge through the system file called *current\_now*. For our target devices we found that this file contains garbage values. Nexus 6 is also equipped with the same fuel gauge and provides correct current readings. This hints that the corresponding systems of Galaxy S3 and S4 were configured to use only ModelGauge.

battery capacity as they age and to understand how the voltage behavior changes while charging as the capacity of a battery decreases. We use the lessons learned to derive a method to estimate FCC of a smartphone battery in the next section.

#### A. Experiment Setup

The measurements are divided in three sets. In the first set, we measured the full charge capacity of Galaxy S3 and S4 batteries. In the second set, we used AC wall and USB2.0 chargers, and kept the smartphones idle in airplane mode in order to maintain a constant current supply to the battery pack. In the second set, we experimented with various system load such as keeping display ON with constant brightness while the devices were in airplane mode. Before charging the batteries, we discharged them by keeping the display ON with a fixed brightness level and then relaxed batteries for five hours. This low rate discharge ensures an empty battery. The fuel gauge manufacturers also conduct their experiments in this way [15]. During the measurements, the room temperature was 21-25° Celsius and each experiment was repeated five times.

From Android BatteryManager we collect battery voltage information, whenever there is a change in the battery level and the timestamp.

#### B. Battery Full Charge Capacity Measurement

We measured the FCC by discharging a fully charged battery. We discharged the batteries of Galaxy S4 and S3 by keeping the devices on in airplane mode at their maximum brightness. While discharging, Yoshoo measures the discharging current and FCC in mAh according to the setup presented in Figure 2(a). The batteries were discharged at 0.272, and 0.256A respectively. The distribution of five FCC measurements for Galaxy S4 and S3 batteries are illustrated in Figure 3. These measured FCCs reveal that the batteries had 3-48% less capacity than their labeled values. Another interesting observation is that the new batteries of Galaxy S3 and S4, B1, have less capacity than the label indicates. There can be two reasons for this; either a battery may actually come with less

capacity, or the battery does not allow itself to be discharged completely and therefore there is always some small amount of charge remain in the battery.

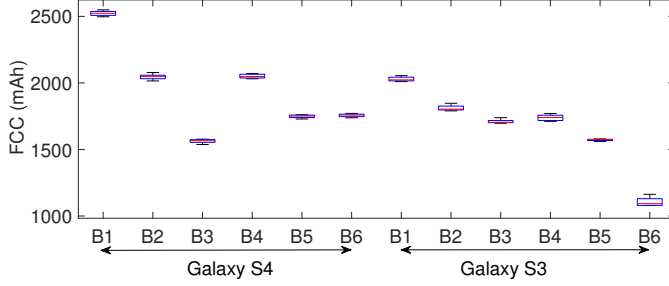


Fig. 3. Measured FCC of the Galaxy S4 and S3 batteries over five runs.

### C. Charging Without System Load

In this case, the smartphones were charged with their standard AC chargers and cables. Figure 4 illustrates the relationship between battery voltage and the SOC while charging new battery, *B1*, on Galaxy S3. The voltage increases sharply within battery level five and then increases gradually over the remaining CC phase. This is because of feeding a constant current to the battery pack during the CC period. After that the voltage remains almost constant during the CV phase as the current is trickled. The SOC level that terminates the CC phase varies with the device model and the corresponding SOC levels are 85 and 76 for Galaxy S3 and S4 respectively. We did not observe any events of SOC fluctuation, i.e. SOC correction, during these charging experiments that were observed while discharging in [7].

Figure 4 further compares the voltage curves of old and new batteries. We see that battery voltage per SOC of the older batteries is higher than that of the new battery. In addition, the voltage of old batteries reach the maximum voltage at lower SOC compared to the new battery. For example, while charging *B6* the CC phase terminates when the SOC reaches 68% compared to that of *B1*. In other words, the older batteries have differing magnitudes less capacity, and the behavior in terms of battery voltage is consistent with the decrease in FCC.

During these charging measurements, we also instrumented the smartphones with Yoshoo as shown in Figure 2(b). Since Yoshoo does not have the functionality to export the measurements, we recorded charging current,  $I_{chg}$ , manually after every 60s and later associated with the SOC update times. Figure 4 shows the measurement results for Galaxy S3 batteries. Note that in some cases the charging current increases from an initial 800 mA to a stable 925 mA at the beginning of the CC phase. Galaxy S4 draws 1560 mA current and exhibits similar charging pattern. Although, the charging current begins to decrease when the voltage reaches its maximum value, the only exception is *B3* for which the current begins to decrease at SOC 73% but the voltage reaches its maximum value when SOC is 80%. This behavior is visible in Figure 4 and persisted across all the measurements with *B3*. In the case of other

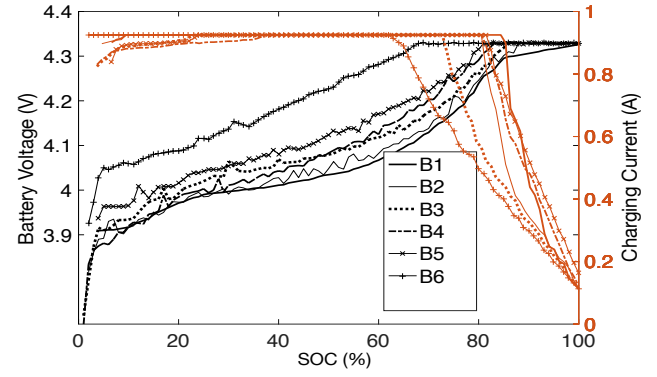


Fig. 4. Battery voltage and charging current vs SOC while charging Galaxy S3 via AC. Voltage curve of a lower FCC battery deviates from the curve of a new battery. The charging current plots (orange color) follow the legend of the corresponding voltage curves.

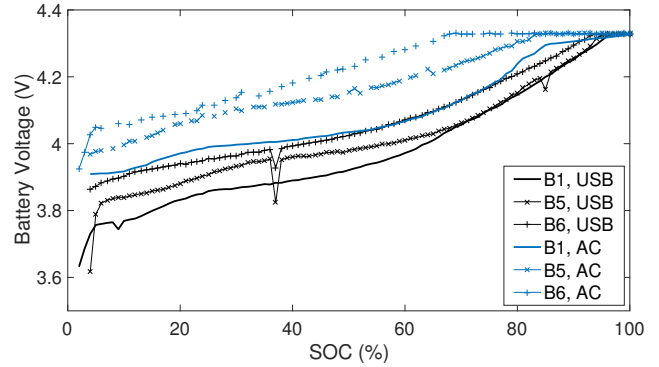


Fig. 5. Battery Voltage behavior of Galaxy S3 while charging via AC and USB2.0.

Galaxy devices, the voltage behavior was in accordance with the charging current. Therefore, the behavior of *B3* is battery specific.

In the case of USB charging, we only experimented with Galaxy S3 and connected the device to a laptop. The device draws 425 mA current from the USB port at 5.00 V. Figure 5 highlights similar voltage behavior as the FCC decreases. However, the battery voltage per SOC is much lower compared with AC charging, as the charging current is lower than the AC charger.

### D. Charging with System Load

Along with the  $I_{chg}$ , we also measured  $I_{bat}$  by placing the Yoshoo power monitor between a device and the battery as shown in Figure 2(c). We measured  $I_{bat}$  while the device was (1) switched off, (2) idle in airplane mode, and (3) actively used.

The measurement results suggest that for the first case,  $I_{bat} \approx I_{chg}$ . For the latter cases,  $I_{bat} = (I_{chg} - I_{sys})$ .  $I_{sys}$  depends on the power consumption characteristics of different hardware components being used. If the system is in airplane

mode, then  $I_{sys}$  varies within 10 mA, which is the standby current consumption of the device in airplane mode. If the device is actively used, then  $I_{bat}$  decreases as the power drawn by different subcomponents, such as display, of the system increases. Consequently, the battery is charged slowly. *Since the battery cannot be charged and discharged at the same time, such a design allows keeping the device on while charging.* However, charging a device with constant system load, such as keeping the display on with fixed brightness, results similar battery voltage behavior to low current USB2.0 charging scenarios shown in Figure 5.

#### E. Summary

The presented voltage curves are of single charging sessions for a battery. The measurement results presented in this section lead us to the following conclusions.

**First,** *The voltage for a particular SOC of an used battery is higher than that of a new battery until the CC phase ends, and the larger the capacity loss, the smaller the SOC level at which the CC phase terminates and the voltage reaches its maximum value.* The reason behind such voltage behavior of aged batteries attributes to their internal resistance, which increases as the battery ages [12].

**Second,** *The charging controller or the charging mechanism does not control the CC-phase charging current as the FCC of a battery decreases.*

**Third,** *The charging controller and the system load dictate the maximum charging current received by the battery.* Therefore, charging a device with a charger of higher output current does not enable fast charging unless both the charging controller and the charger support quick or fast charging.

**Fourth,** *Mobile devices provide more reliable SOC estimates while charging compared to the discharging scenarios presented in [7].* The reason for such better performance is that there is always some incoming charge from the charger.

### IV. FULL CHARGE CAPACITY MODELING

The state-of-art approach used by modern smartphones to estimate FCC is the number of charging cycles [17]. Modern fuel gauges use Coulomb counter to measure the FCC of a battery. In this section, we propose a new FCC estimation technique based on the charging current and voltage behavior while charging.

#### A. Full Charge Capacity Modeling

As C-rate is the ratio of charging current and FCC, it is possible to compute the present FCC of a battery from the C-rate. The equation to compute C-rate from the battery initial capacity and charging current is the following.

$$C_{new} = \frac{I_{bat}}{FCC_{new}} \quad (1)$$

In Figure 4, we have shown that the length of CC phase reduces as the capacity of the battery reduces. However, the

charging current (mA) drawn from the charger during the CC phase does not change as the FCC of the battery decreases. Therefore, the  $FCC_{now}$  can be defined as

$$C_{now} = \frac{I_{bat}}{FCC_{now}} \quad (2)$$

Therefore, (1) and (2) together can be expressed as

$$FCC_{now} \times C_{now} = FCC_{new} \times C_{new} \quad (3)$$

$$\frac{FCC_{now}}{FCC_{new}} = \frac{C_{new}}{C_{now}} \quad (4)$$

And consequently, the present capacity of the battery can be computed with eq. (3). This reveals that the present capacity of a battery is the ratio of the charging C-rates with present unknown capacity and the new battery. This is shown in equation eq. (4). In the above equations, the C-rate of a new battery,  $C_{new}$ , can be derived from the battery capacity and charging current information. The batteries usually come with labeled capacity values. The charging C-rates of Galaxy S3 and S4 are 0.44 (925/2100) and 0.6 (1560/2600) C respectively while charging new batteries via their AC chargers.

#### B. Extending Model with C-rate from SOC Updates

Given the initial battery capacity and the charging current,  $FCC_{now}$  and  $C_{now}$  are two unknown variables, which depend on each other. Therefore, we need to find the  $C_{now}$  in order to compute the  $FCC_{now}$ . In this section, we devise a method to estimate  $C_{now}$ .

The definition of CC-CV charging algorithm states that the charging current remains constant during CC period and CV phase begins when the battery voltage reaches its predefined maximum value. Therefore, we can estimate the rates from the time stamp of the same SOC updates for the charging measurements presented in Section III. The equation is the following,

$$C_{SOC_{i \rightarrow n}} = \frac{36 \times (SOC_{i+n} - SOC_i)}{t_{SOC_{i+n}} - t_{SOC_i}} \quad (5)$$

where 36 is the time in seconds to charge one percent at 1 C-rate.

Using eq. (5), we can estimate the cumulative charging rate over the CC period of charging. Consequently, eq. (5) can be expressed as the following.

$$C_{SOC_{CC}} = \frac{36 \times \Delta SOC_{CC}}{\Delta t_{CC}}, \quad (6)$$

where  $\Delta SOC_{CC}$  is the length of CC phase and  $\Delta t$  is the time to charge the battery till CC phase. Finally, eq. (4) can be further expressed which is equivalent to the ratio of average one percent charging times.

$$\frac{FCC_{now}}{FCC_{new}} = \frac{\Delta SOC_{cc-new}}{\Delta t_{cc-new}} \times \frac{\Delta t_{cc-now}}{\Delta SOC_{cc-now}} \quad (7)$$



TABLE I. AVERAGE FCC MEASURES FROM FIVE ROUNDS OF MEASUREMENTS AND COMPARISON WITH FCC ESTIMATES FROM AC AND USB2.0 CHARGING C-RATES.

Smartphone Model	$FCC_{now}$ (loss %)	$C_{new}, C_{now}$ (AC)	$C_{new}, C_{now}$ (USB2.0)	$FCC_{now}$ (AC, USB2.0) Model
GS4 (B1)	2522 (3%)	0.60, 0.62	0.164, 0.168	2516, 2538
GS4 (B2)	2046(21%)	0.60, 0.76	0.164, 0.207	2052, 2059
GS4 (B3)	1562(40%)	0.60, 1.0	0.164, 0.272	1560, 1568
GS4 (B4)	2050(21%)	0.60, 0.76	0.164, 0.207	2052, 2090
GS4 (B5)	1748(33%)	0.60, 0.89	0.164, 0.243	1753, 1755
GS4 (B6)	1754(32%)	0.60, 0.89	0.164, 0.242	1753, 1762
GS3 (B1)	2028 (3%)	0.44, 0.46	0.202, 0.209	2008, 2030
GS3 (B2)	1811(14%)	0.44, 0.51	0.202, 0.234	1812, 1813
GS3 (B3)	1710(19%)	0.44, 0.54	0.202, 0.248	1711, 1710
GS3 (B4)	1738(17%)	0.44, 0.53	0.202, 0.244	1743, 1738
GS3 (B5)	1511(25%)	0.44, 0.61	0.202, 0.281	1515, 1510
GS3 (B6)	1106(48%)	0.44, 0.84	0.202, 0.384	1100, 1104

## V. FCC ESTIMATION MODEL VALIDATION

In this section, we validate the model estimates by comparing with the measured FCCs of the Galaxy S3 and S4 batteries. First, we validate the FCC estimates, when the C-rate is derived from the ratio of the charging current and FCC. Next, we validate the estimates when C-rate is derived from SOC updates. This section further emphasizes the performance of the model with a number of practical scenarios, such as charging with different chargers, partial charging, ambient temperature, and a device with Coulomb counter-based SOC estimation technique.

### A. FCC estimate from Charging Current C-Rate

The distribution of five FCC measurements for Galaxy S3 and S4 batteries are illustrated in Figure 3 and the averages are presented in Table I. We next estimate the charging C-rates during the CC-phase according to eq. (2) for both AC and USB charging and present in the table. These estimates highlight that charging C-rate increases as the FCC decreases. Therefore, the relation can be stated as the following.

*Proposition 1:* If the charging current is constant, the charging C-rate increases as the FCC of a battery decreases.

Finally, we estimate capacity using eq. (4) and compare with the measurement results in the table. The table shows that the model estimates FCC of the batteries quite reasonably for both the AC and USB charging.

### B. FCC estimate from Charging SOC C-Rate

Although eq. (7) shows that  $FCC_{now}$  can be estimated from the average charging time per SOC of a new and old battery, we use cumulative charging C-rates, according to eq. (5), computed from the SOC updates collected during the charging measurements presented in Section III-C. The reason is that C-rate further enables to estimate the charging current and to detect battery change (see Section VII). Besides, it is the widely accepted metric in battery capacity and SOC modeling.

We plot the C-rates for every SOC from a single charging session in Figure 6. According to the figure, as the estimated C-rates are not constant during the CC phase, it is a challenge to

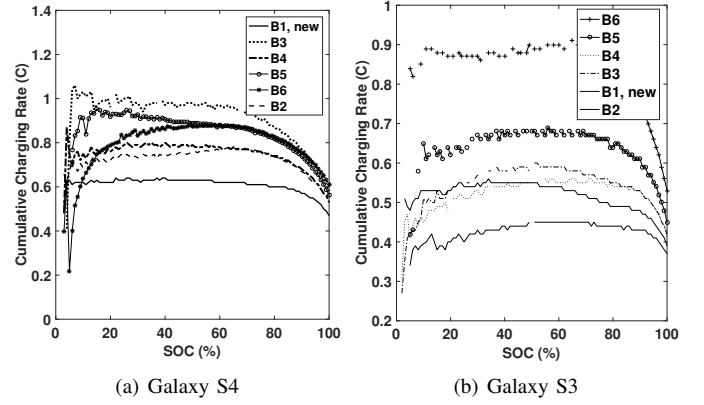


Fig. 6. Example C-rate curves of the Galaxy devices with new and long used batteries for AC charging.

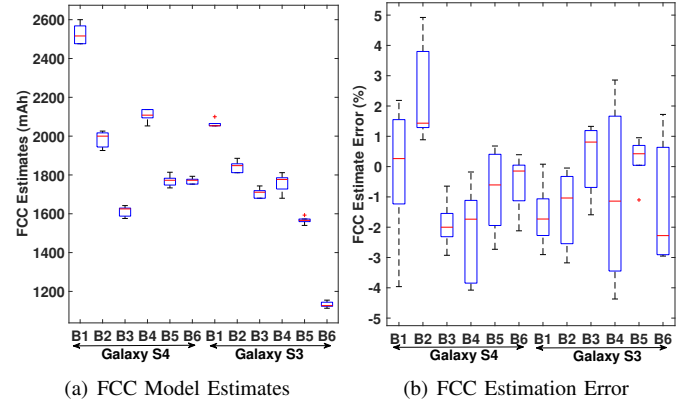


Fig. 7. FCC estimates of the Galaxy S4 and S3 batteries and the estimation error while charging via AC.

select the range of SOC values over which the C-rate should be calculated given the rate curves. For instance, although the CV phase of Galaxy S3 with battery B2 (Figure 6(b)) starts when the SOC is 80%, we notice that the battery is charged with the maximum C-rate till 40% and after that C-rate gradually decreases. Multiple measurements confirmed that this behavior is battery specific. Because of this, we explore and validate two different options; 1) we select the range of SOC values that cover the whole CC phase or 2) we select the SOC range that yields the highest C-rate (we call this max C-rate). The lengths of the CC phase are derived from the voltage curve of the battery.

If we take the rates at the point where the CC phase ends, we find that AC charging rates of the new batteries of Galaxy S3 and S4 are 0.44 and 0.59 C respectively. They are very close to the measured rates presented in Table I. Figure 6 also highlights that although the charging rate from the wall charger is almost constant during the CC phase irrespective of the battery capacity, the C-rates of the older batteries are higher than the new batteries. Figure 7(a) shows the model estimates from the C-rates at  $SOC_{CC}$  and Figure 7(b) demonstrates that these estimates are close to the measured values and they suffer only  $\pm 5\%$  error. In Figure 8(a) and 8(b), we further notice

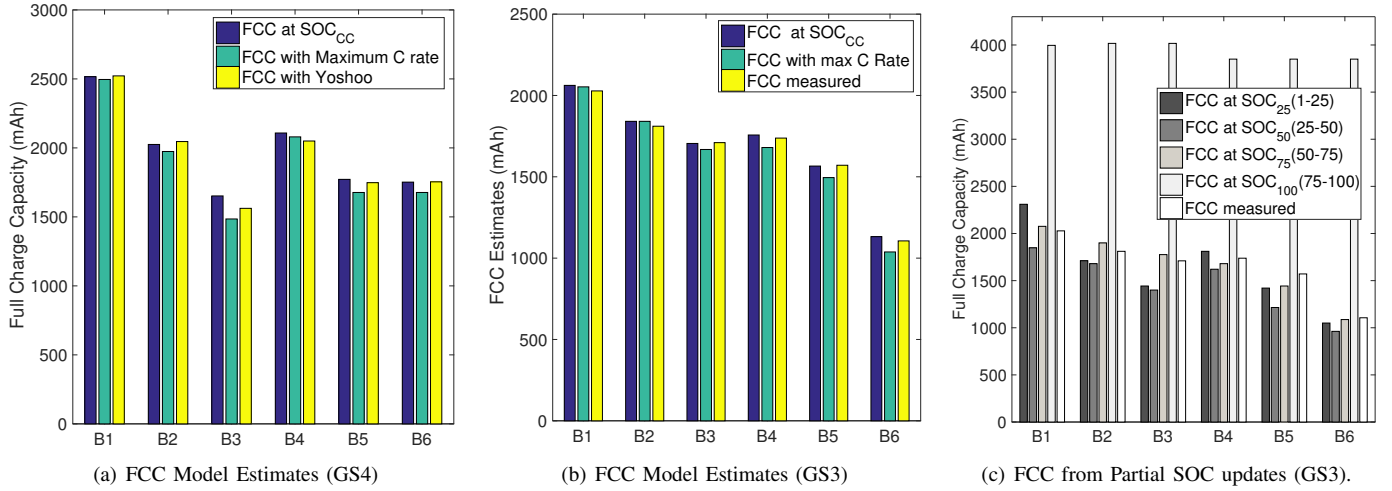


Fig. 8. Comparing FCC estimates from different C-rates with the measured FCCs. Figure 8(c) compares the FCC estimates with when C-rates are derived from partial SOC updates.

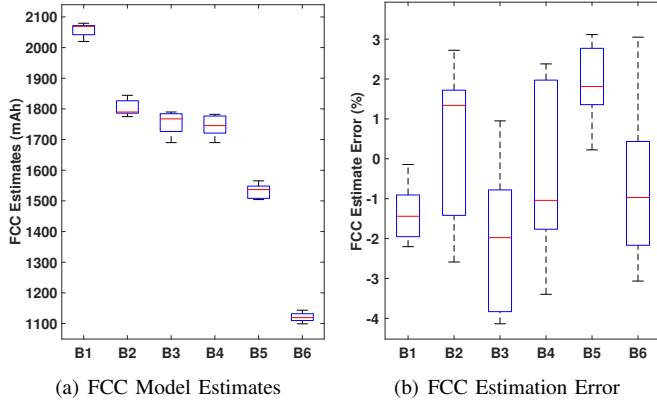


Fig. 9. FCC estimates of Galaxy S3 batteries while charging via USB2.0.

that both rate selection approaches estimate  $FCC_{now}$  with less than 10% error. The C-rate at  $SOC_{CC}$  yields FCC estimates that are closer to the measurements more often than the other approach.

#### C. FCC estimate from Partial SOC updates

Although the Android BatteryManager broadcasts SOC update events regularly, there may be only few updates available in practice. The underlying reason can be device-specific behavior in reporting SOC updates or that the device is in doze mode and unable to broadcast these events. In addition, a user may charge the device when the device is off and switch it on only after the device is charged to a reasonable capacity. Therefore, our method has to work also when it has only partial SOC updates available. Figure 8(c) illustrates the FCC estimates at the boundaries of four different SOC intervals and compares with the measurement results. We notice that within 75% SOC boundaries the FCC estimates are close to the measurement results. Beyond that SOC level, the error

increases significantly due to the trickling charging current during the CV phase.

#### D. FCC estimate with Different Chargers

A device can be charged with different chargers connected via different cables. We further estimate the FCC of Galaxy S3 batteries from the C-rates while charging via USB. Figure 9(a) demonstrates similar estimates as AC charging and the corresponding estimation error is within  $\pm 5\%$  (see Figure 9(b)).

These results from AC and USB charging also highlight that the ratio of USB and AC rates always provide similar capacity estimates of a battery. A constant system load, such as keeping the display ON with a fixed brightness or the system in airplane mode, while charging ensures constant charging current for the battery and thus produces similar results. Table I shows similar estimates for both devices, when C-rate is estimated from charging current. Therefore, the relation between C-rate and FCC can be further extended as the following.

*Proposition 2:* As long as the charging current is constant at any arbitrary value, the ratio of the corresponding C-rates of a new and an old battery will always indicate the present FCC of the old battery.

We later in Section VII discuss further how this relation can be applied to find the charging current of any unknown charger.

#### E. FCC at Low Temperature

We further conducted charging experiments with varying temperatures. Figure 10(a) demonstrates voltage behavior of the new Galaxy S3 battery, B1, at different temperatures while charging. The voltage behaves in a similar fashion to those of the old batteries. Figure 10(b) shows that the FCC of B1 and B4 decrease when they are charged at lower than room temperature. In addition, we investigated whether the effect of such charging has short or long-term effect on FCC. We discharged the batteries at room temperature, after



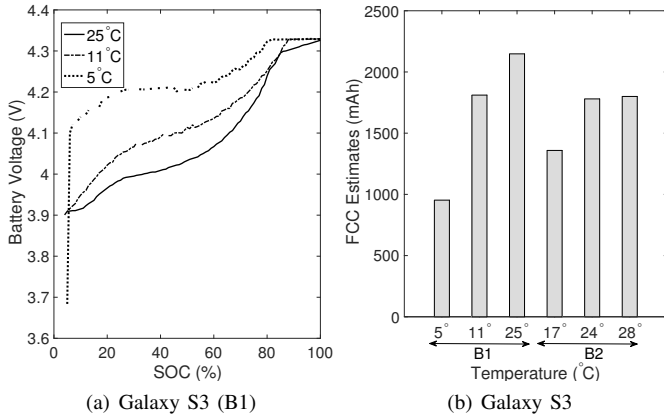


Fig. 10. The effect of temperature on battery voltage and FCC estimates while charging via AC.

we had charged them at lower temperature. Afterwards, we relaxed them for five hours, and then charged again at room temperature. We found that all the batteries retained their earlier capacity state. Such results suggest that temporary low temperature charging has short-term effect on FCC of the battery.

#### F. FCC of a Coulomb Counting Device

In the earlier sections, we measured and estimated FCC for the devices with voltage-based SOC estimation techniques. We also considered a Coulomb counter-based Nexus 6 device with labeled FCC of 3220 mAh. Unlike the Galaxy devices, Nexus 6 has a functional Coulomb counter [16]. In this case, we modified Carat to retrieve charging current with Android `BATTERY_PROPERTY_CURRENT_NOW` API at one second interval. Similar to the Galaxy devices, the charging current remains almost constant during the CC phase. The charging rates of the device with USB2.0, standard 5 V AC, and turbo 9 V AC charging are 0.142(457/3220), 0.43(1391/3220), and 0.74(2380/3220) C respectively.

The CC phases with the corresponding chargers terminate at 95, 80, and 70 SOC and the charging SOC rates are 0.146, 0.44, and 0.75 C respectively. These estimated C-rates are approximately equivalent to the charging current rates and provide battery capacity 3131, 3146, and 3177 mAh respectively that are very close to the labeled capacity. At the same time, Carat also measured FCC by reading `BATTERY_PROPERTY_CHARGE_COUNTER` when the battery was full. The average capacity reported was 3157 mAh. Therefore, the estimation error compared with the Coulomb counter measures is within  $\pm 5\%$ .

#### G. Discussion

We have demonstrated that the relative charging rate within the CC phase of a battery increases as the FCC decreases. Based on this C-rate behavior, we have proposed an FCC estimation model. The models can be enforced with various constant current chargers and even with partial charging. The accuracy of the approach is above 90%.

The approach equally works with C-rates derived from the charging current and SOC updates. Among the rate selection methods, described in Section V-B, we select the C-rate at  $SOC_{CC}$ . The method suffers from similar errors with voltage and Coulomb counter-based devices. We apply the method with the devices in the crowd-sourced Carat data set and compare with the C-rate of the new batteries of Galaxy devices. We have found that 43-50% of the devices of these models suffered from capacity loss and a significant number of them lost 25% of the capacity. We describe Carat data set more detail and the crowd-sourced FCC estimation method in Section VI.

## VI. CROWD-SOURCED BATTERY ANALYTICS

Smartphones typically do not report the capacity of a battery and the charging current. Besides, a user may use a different charger rather than the stock charger. Therefore, in this section we devise a statistical method to find the  $FCC_{now}$  of an unknown device given a large collection of the Android BatteryManager data of a particular smartphone model. This also enables online collaborative FCC and charging debugging of smartphone batteries.

### A. Data Set & Pre-processing Charging Samples

The Carat application collects different information from smartphones as samples whenever there is a change in the SOC or battery level. Among various information available in a sample, we only consider the time when the sample was taken, the SOC, battery voltage, battery health, the type of charger, and the CPU usage. Hence, the reduced sample looks as  $S = (t, (SOC : i\%), (voltage : V), (temp : C) (charger : ac/usb), (health : good/dead/cold), (cpu : x\%))$ . We next consider the samples with the *charger* attribute of “ac” and health attribute of “good” value. As it is demonstrated earlier that charging a battery at lower than room temperature can affect the FCC, we consider only those samples with battery temperature reporting 21-40°C, as the capacity variation within this range is very small. There were about 3 million charging samples and about 22 K devices of 1200 models had more than 5 good charging samples. However, the samples did not have the display status, i.e., ON/OFF, of the devices during the sample collection.

We next sort the good AC charging samples of a user according to the time stamp in order to find the charging time between two consecutive samples. First, we group the samples that belong to same charging events. Ideally, a charging event begins when a charger is plugged in and ends when the charger is unplugged. However, the construction of the events in this way is difficult from the data set as a user may power on or off the phone while charging and power on when the battery is charged to a reasonable capacity. The charging algorithms terminate charging once the charging rate falls to 0.07 [10]. Therefore a mobile device spends at most  $\frac{36}{0.07} = 514$  seconds to charge one percent. We next add this derived attribute in the samples and finally we obtain the following kind of samples:  $S = (t, (SOC : i\%), (voltage : V), (\Delta t : S), (cpu : x\%))$ . All the pre-processing is done using Spark [18] with 7 machines each having 8 CPU cores and 30GB RAM.

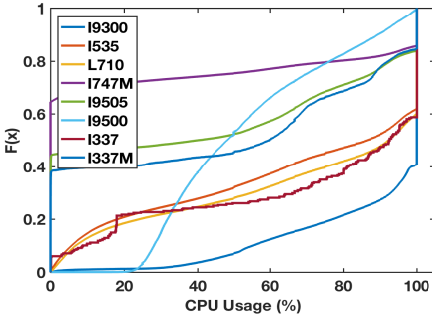


Fig. 11. Distribution of CPU usages reported in the samples of some popular smartphone models. More than 50% of the samples from every model had 100% CPU utilization.

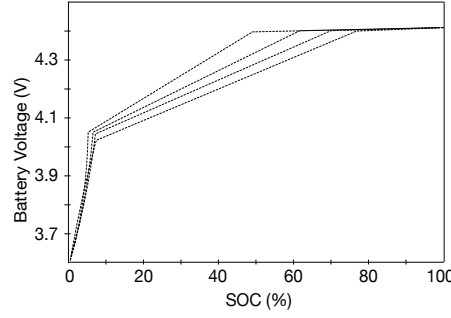


Fig. 12. The voltage curve can be divided into three linear segments. The slope of the second segment is lower than the first segment and the third segment is parallel to the X-axis.

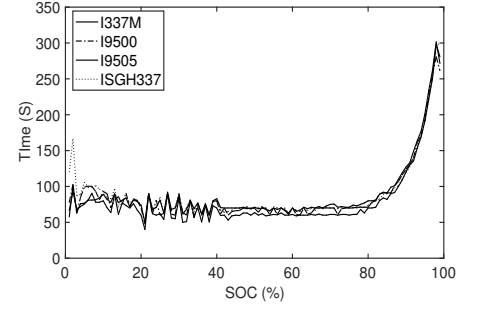


Fig. 13. One percent charging time curves of various Galaxy S4 models. One percent charging time is almost constant with respect to SOC within the CC phase (75%).

## B. Methodology

Our method relies on a number of steps. First, we find the model-specific voltage curve, the length of the CC phase and the charging C-rate of a smartphone model corresponding to the standard AC charger. We next find the user-specific voltage and charging rate during the CC phase from the latest samples. These model- and user-specific charging rates are equivalent to the  $C_{new}$  &  $C_{now}$ , respectively. Finally, we compare these two rates according to (4) to determine the capacity loss.

Determining model-specific voltage and rate curves from the crowd-sourced data is not trivial. There are a number of challenges. If there is system load while charging, battery voltage will increase slowly and it will take longer to charge. The data set had more than two-third samples with more than 75% CPU utilization. The distribution of CPU utilization of some popular models, presented in Figure 11, indicates that these devices had significant system load while charging. Further, it is difficult to identify whether the standard charger of a device model had been used to charge a device or not. The consequence of these factors is slower charging and gradual increase of battery voltage. Carat is an energy debugging engine, the users can also be biased towards having devices with the lower FCCs and thus can have faster charging time and voltage increase.

Earlier, in Section III, we emphasized that battery voltage can characterize both kinds of biasness. Therefore, we apply *G-test* [19] on per battery level or SOC-specific voltage distribution to determine the skewness (see *Alg.1*). Left skew implies biasness towards lower FCC devices. If the distribution is right skew and symmetric then the distribution is affected by the samples with device usage or substandard AC charger. Therefore, if the distribution of voltage for an individual SOC is left skewed, we consider the median voltage, else we consider the 75th percentile as stated in *Alg.1*. Such selection is based on voltage behavior according to system load, charger and FCC as presented in Section III, and the distribution of CPU load present in the dataset.

## C. Reference Voltage Curves

The reference voltage curve constructed from the crowd-sourced data should resemble the curve of a new battery.

### Algorithm 1 Reference Voltage Curves and C-rates

```

1: function MODELCURVES( model )
2:   for each soc ∈ SOC do
3:     volts=getVoltages(soc)
4:     times=getTimes(soc)
5:     sk=GTest(0.05, volts)
6:     if (skew==right) || (skew==symm) then
7:       [index, volt]=sevenfive(volts)
8:     else if (skew==left)
9:       [index, volt]=median(volts)
10:    end if
11:    voltage(soc)=volt
12:    time(soc)=times(index).time
13:  end for
14:  [mVCrv, mcc]=estimate(voltage)
15:  mVCrv=delOutliers(mVCrv(mVCrv))
16:  [mVCrv, mcc]=estimate(mVCrv)
17:  mTCrv=delOutliers(mTCrv(1:mcc))
18:  mTCrv=estimate(mTCrv(1:mcc))
19:  mRate=mRateC(mTCrv(1:mcc))  ▷ C-rate eq. (6).
20:  return mRate
21: end function

```

Therefore, the number of samples per model should be sufficient and it is essential to have as many non-null SOC elements as possible in the curve. We select the device models, which had a minimum 250 samples. This gives us a wide coverage of different models. Among 370 models, we found that approximately 300 models had more than 90% non-null SOC entries in their reference curves. The remaining device models had more than 60% SOC entries per curve.

However, the 250 samples do not guarantee that there will not be any non-null SOC entry. Figure 12 illustrates that the charging voltage curves can be split into three linear segments. The length of first segment is approximately ten SOC. The lengths of the second and third segments vary, which depend on the capacity of the battery. As the FCC decreases, the length of the second segment decreases and the third segment increases. The second segment has a positive slope, whereas the third segment is parallel to X-axis. We use linear interpolation/extrapolation to estimate the missing values

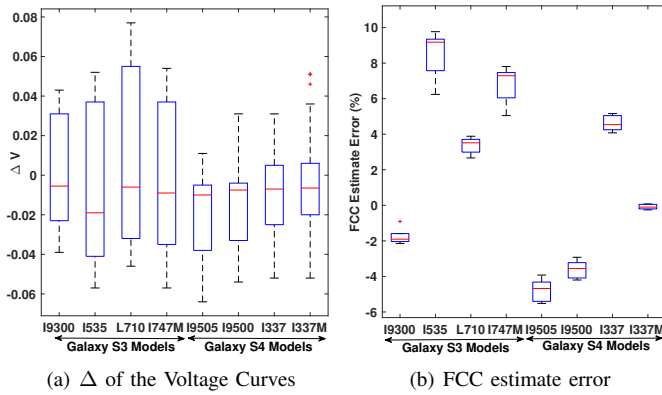


Fig. 14. **The performance of the crowd-sourced approach.** The plots present the difference of battery voltage per SOC between the crowd-sourced and the measured curves and the errors compared to the measured FCC new batteries, B1, of Galaxy S3 and S4.

in the reference voltage curves (line 14, Alg.1).

Neither the 250 samples nor the estimation techniques guarantees that the voltage for a particular SOC in the reference curve is not an outlier. Therefore, we first find the absolute voltage difference for two consecutive SOC. This list of differences is a normal distribution and we next apply iterative *Grubbs test* on this distribution to detect the outliers in the voltage curve using the Matlab function presented in [20] (line 15, Alg.1). Grubbs test determines whether the tested value is the highest/lowest and furthest from the sample mean [21]. We again estimate the outliers.

#### D. Reference C-rates

Similar to the voltage curves, we also construct model-specific rate curves. Towards this, we first select the one percent charging time curve constructed from the reference voltage curve samples in Section VI-B. Similar the voltage curve, a charging time curve is required to have charging time for all the SOC within the CC phase boundary, which is equivalent to the first two segments of the voltage curve. As shown in Figure 13, these curves are almost parallel to X-axis within the SOC boundary. Naturally, the corresponding time curve also may contain outliers. The figure shows that the charging time within the SOC boundary follows normal distribution and thus we apply the *Grubbs test* to find the outliers. Finally, we use linear interpolation/extrapolation to replace those outliers (line 17, 18, Alg.1).

Once we have the charging time curve of a model, we apply eq. (6) to obtain the cumulative C-rate at CC phase SOC boundary. The third column in Table II presents the C-rates of different smartphone models computed from the data set according to Alg.1.

#### E. Crowdsourcing Method Validation

1) *Crowd-sourced Voltage Curves:* The validation of the crowd-sourced method for determining the voltage curve depends on the length of the CC-phase and how close the curve is to that of a new battery.

We find the CC phase length from the reference voltage curves. It is the SOC value when the voltage reaches the maximum  $4.2/4.35 \pm 0.05V$  (i.e. *mcc* in Alg.1). We have identified SCH-I535, SPH-L710, SGH-I747M, GT-I9300 are Galaxy S3 models. In Table II, we notice that the CC-phases are spread over 79-83 SOC that are very close to the measured value for the new Galaxy S3 battery, B1, presented in Section III. Similarly, the Galaxy S4 models, such as SCH-I545, SGH-I9500, SGH-I337, SGH-I337M and GT-I9505 have CC-phases that are approximately 75 SOC which is equivalent to the measured CC phase of B1.

Figure 14(a) compares the voltage curves from the measurement and crowd-sourced data of some popular Galaxy S3 and S4 models. The difference of voltage per SOC is computed by subtracting the crowd-sourced voltage curve from the measured curve. It is shown that the distances are very small and the median distance is close to zero. This implies that the crowd-sourced voltage curves for our experimental models are very close to those of new batteries presented in section III-C.

2) *Crowd-sourced C-rates and FCC Estimates:* We further investigate the effectiveness of our crowd-sourced rate estimation technique. Among the models presented in Table II, six models are of Galaxy S3 and five are of Galaxy S4. We know their  $FCC_{new}$  and charging current. We first estimate the FCC of the experimental batteries with the crowd C-rate of the Galaxy S3 and S4 models, and then compare with the measured FCCs of new batteries, B1, presented in Figure 3. Each boxplot in Figure 14(b) represents the error of the estimates compared with the measured FCCs of B1. We notice that the error is less than 10% for both device models. The errors for other device models in table II are also in the same range as the differences in C-rates are within  $\pm 0.05 C$ . The negative error indicates capacity is over-estimated, whereas the positive error indicates under-estimation. The table also highlights that the C-rates for most of the models are lower than the measured values. This further indicates the influence of device utilization on C-rate.

From Internet, we have also collected the capacity and the charging current of other models. We notice that the crowd-sourced reference rates are within  $\pm 0.05 C$  of their computed values for these popular models.

#### F. User Rate Curves and FCC in the Wild

In order to find the reduced FCC of a single user device, we compare the model-specific reference rate with the rate from a user-specific charging time curve. As our interest is the latest battery capacity of a device, we construct one percent charging time curve from the corresponding voltage curve samples of the latest month. We select the maximum voltage per SOC, as it guarantees less device utilization and the recent state of the battery at the same time. Again, it is important to have adequate number of samples for a user curve as well. We consider only those devices, which reported at least 25 samples within the SOC boundary of the second segment. Similar to the model reference curves, we also detect outliers and apply linear interpolation for estimating the missing values in the

TABLE II. TOP SEVENTEEN DEVICE MODELS AND THEIR BATTERY PROPERTIES. THE COLUMNS REPRESENT THE CC PHASE LENGTH IN SOC, REFERENCE C-RATES DERIVED FROM THE CROWD, THE C-RATES COMPUTED FROM THE RATIO OF CHARGING CURRENT (mA) AND THE BATTERY CAPACITY (mAh), USERS WITH MORE THAN 25 SAMPLES WITHIN THE CC-PHASE, AND THE PERCENTAGE OF THE USERS WITH LOWER FCCs.

Device Model	CC	Crowd C-rate	C-rate $\frac{mA}{mAh}$	Users	Poor FCC
M1-GT-I9100 (GS2)	76	0.35	$0.39, \frac{645}{1650}$	420	35%
M2-GT-I9300 (GS3)	83	0.45	$0.44, \frac{925}{2100}$	284	42%
M3-SGH-T999	80	0.43	$0.44, \frac{925}{2100}$	284	52%
M4-SGH-I747	80	0.42	$0.44, \frac{925}{2100}$	524	56%
M5-SGH-I747M	80	0.40	$0.44, \frac{925}{2100}$	232	58%
M6-SPH-L710	79	0.42	$0.44, \frac{925}{2100}$	442	60%
M7-SCH-I535	81	0.39	$0.44, \frac{925}{2100}$	619	59%
M8-GT-I9505 (GS4)	75	0.64	$0.60, \frac{1560}{2600}$	317	27%
M9-GT-I9500	75	0.63	$0.60, \frac{1560}{2600}$	78	24%
M10-SCH-I545	74	0.57	$0.60, \frac{1560}{2600}$	167	36%
M11-SGH-I337	75	0.62	$0.60, \frac{1560}{2600}$	135	43%
M12-SGH-I337M	75	0.60	$0.60, \frac{1560}{2600}$	34	50%
M13-HTC One	87	0.30	$0.33, \frac{750}{2300}$	253	51%
M14-Galaxy Nexus	74	0.59	$0.57, \frac{1000}{1750}$	341	23%
M15-Nexus 4	84	0.40	$0.44, \frac{925}{2100}$	920	47%
M16-Nexus 5	91	0.55	$0.59, \frac{1350}{2300}$	252	20%
M17-Nexus 7	91	0.27	$0.27, \frac{1150}{4325}$	532	42%

#### Algorithm 2 Users' Battery FCC Loss Detect and Estimate

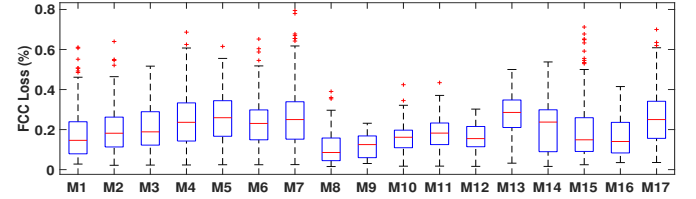
```

1: function BATTERYFCCESTIMATE
2:   for each model  $\in$  MODELS do
3:     mRate=ModelRateC(model)
4:     for each device  $\in$  modelDEVICES do
5:       ▷ uCrv is constructed from the latest one month samples
         reporting the maximum voltage reported per SOC.
6:       [uVcrv,ucc]=interpolateV(uVCrv)
7:       uTCrv=uTimeCurveVoltage(1:ucc)
8:       uRate= uRateC(uTCrv(1:ucc))    ▷ eq. (6).
9:        $\Delta r = uRate - mRate$ 
10:      if ( $\Delta r > 0$ ) && ( $ucc > 0$ ) && ( $ucc \leq mcc$ ) then
11:        uCap = mRate/uRate
12:        uLoss=(1-uCap)
13:      end if
14:    end for
15:  end for
16: end function

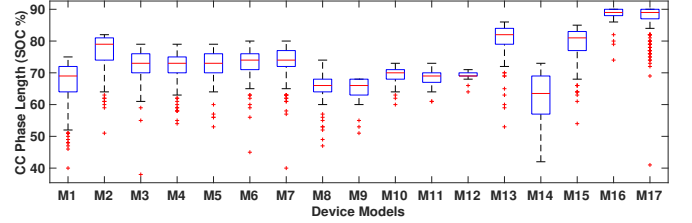
```

user-specific curves. Once, we have the user C-rate at the CC phase SOC boundary (line 8, Alg.2), we compute the capacity loss using the model.

Among 9560 user devices, 3800 devices of 333 models had reduced battery capacity. The sixth column in Table II shows that more than 50% of the devices of seven popular models had reduced battery capacity. The ratio of such users is the lowest with Galaxy S4 (GT-I9505, GT-I9500), Galaxy Nexus, and Nexus 5 models. The range of FCC losses for these users are illustrated as boxplots in Figure 15(a). We can see that most of the devices of these models had less than 20% capacity loss.



(a) FCC loss distribution of the devices.



(b) CC-phase lengths of the devices with reduced capacity.

Fig. 15. Battery properties of user devices of the smartphone models presented in Table II.

We further looked into the length of CC phase of user devices of these models and present in Figure 15(b). Galaxy S3 devices suffered a median of 20% FCC loss and their median CC phases 75 SOC. The Galaxy S4 devices had less 20% capacity loss and the corresponding median CC phases are within 65-70 SOC. Note that the length of CC phases can be zero and we have found that around 38 devices among all the devices had CC phase length of zero. In other words, these devices had significant capacity loss.

#### G. Summary

The presented method in Section VI-F to estimate FCC loss of a user device always strives for the maximum battery voltage per SOC which also guarantees maximum C-rate. The method assumes that the users charged their devices with the stock AC charger. Therefore, it undermines the fact that a user may charge a device with a charger that delivers lower charging current compared to a standard charger of the device model. Although the device models presented in Table II do not support Quick or fast charging, still some older device models may accept Quick charging and thus higher charging rate. In these cases, comparing the user C-rate with the model C-rate would underestimate the FCC loss. Therefore, the amount of FCC loss of a user presented in Figure 15(a) is either under or firmly estimated.

### VII. BATTERYSENSE

The crowd-sourced approach suffers from two competing sources of bias that affect the reference voltage and rate curves; the samples gathered during active usage of the device and the samples from the lower FCC devices. In addition, it is impossible to infer the charger configuration of a user device and the charging C-rate of the user device. Therefore, we propose an approach, called BatterySense, to be implemented in Android mobile devices to estimate FCC, validate device

---

**Algorithm 3** FCC Loss and Abnormal Charging
 

---

```

1: function BATTERYSENSE
2:    $I_{usb} = 425$   $\triangleright$  USB 2.0 port output current 425mA.
3:    $C_{usbnow} = socCrate(usb)$   $\triangleright$  C-rate eq.(5).
4:    $FCC_{now} = \frac{I_{usb}}{C_{usbnow}}$   $\triangleright$  FCC according to (2) in mAh.
5:    $C_{crowd} = crowdCrate(ac)$   $\triangleright$  Carat back-end.
6:    $\triangleright$  Apply eq. (4).
7:    $FCC_{new} = (FCC_{now} \times C_{acnow}) / C_{crowd}$ 
8:    $\triangleright$  FCC Loss in mAh.
9:    $FCC_{loss} = (FCC_{new} - FCC_{now}) / 100$ 
10:  if  $FCC_{loss} < -5\%$  then
11:     $notify(new/extra, FCC_{now})$ 
12:  else
13:     $notify(loss, FCC_{now})$ 
14:  end if
15:   $C_{acnow} = socCrate(ac)$   $\triangleright$  C-rate eq. (6).
16:   $I_{acbat} = FCC_{now} \times C_{acnow}$ 
17:   $C_{acnew} = I_{acbat} / FCC_{new}$ 
18:  if  $(C_{crowd} \approx C_{acnew})$  then
19:     $notify(okay)$   $\triangleright$  Stock AC Charger.
20:  end if
21:  if  $(C_{crowd} >> C_{acnew})$  then
22:     $notify(slow)$   $\triangleright$  Charger or Cable issue.
23:  end if
24:  if  $(C_{crowd} << C_{acnew})$  then
25:     $notify(fast)$   $\triangleright$  Fast Charging.
26:  end if
27:   $\triangleright$  Updating Carat Back-end.
28:   $update(deviceID, C_{acnew}, I_{acbat})$ 
29:   $update(deviceID, usbSamples, acSamples)$ 
30: end function

```

---

and crowd-sourced model-specific C-rates. The approach can also detect abnormal charging behavior.

We first demonstrate how to find the present battery capacity of a device locally and then the AC charger configuration. Next, we demonstrate how to combine the crowd C-rate and locally computed C-rate to find the charging efficiency. All these mechanisms are incorporated in BatterySense.

The steps are presented in *Alg.3*. Unless a device uses a Coulomb counter-based SOC estimation approach, it is impossible to measure FCC and charging current. Nevertheless, the charging current from the USB2.0 ports of computer is approximately 425 mA at 5.0 V. Therefore, BatterySense first constructs voltage curve and computes C-rate for USB2.0 charging from unbiased samples, i.e., when the device is completely idle. Then, it estimates  $FCC_{now}$  of the device. As for example, the USB C-rate for B3 of Galaxy S3 is 0.248 C and thus,  $FCC_{now} = \frac{I_{usb}}{C_{now}} = 1713$  mAh.

BatterySense next requests  $C_{new}$  or model C-rate to Carat back-end for AC charger, which is derived according to the statistical approach described in Section VI. This is used to find the FCC loss of the battery. However, it is common that users replace their standard batteries with substandard ones with higher capacity. In such cases FCC loss will be negative and therefore, the presence of such batteries also can be detected with this crowd-sourced C-rate.

BatterySense computes C-rate for AC charging locally. As the battery is charged immediately, it can be considered

that AC charging C-rate,  $C_{acnow}$ , would also provide similar  $FCC_{now}$  estimate. From these two variables, BatterySense finds the AC charging current for the battery  $I_{acbat}$  in mA. For instance, the  $C_{acnow}$  of B3 is 0.54 C and this provides the charging current drawn by the device from the charger,  $I_{acbat} = 1713 \times 0.54 = 925$  mA. This is very close to the measured AC current drawn by the Galaxy S3 device while charging via AC.

BatterySense further computes  $C_{acnew}$  and compares with the crowd C-rate. If the difference between these two is significant, the device is being charged either abnormally slow or fast. Consequently, BatterySense notifies user about the charging quality as well. In addition, BatterySense updates Carat-backend with the locally computed C-rate,  $C_{acnew}$ , and charging current of the corresponding charger. This further enables the back-end to refine the crowd C-rate.

## VIII. RELATED WORK

Battery capacity usually is modeled as a function of charging cycle and temperature [39]. A number of data-driven strategies exist that predict the capacity as the battery ages and the number of charging cycle increases. Yin et al. [37] and Liu et al. [38] applied a few variations of the Gaussian process regression to predict the capacity as a function of charging cycle. Guo et al. [40] proposed a time dependent nonlinear least square method to estimate battery capacity while charging. The coefficients of the model are taken from an equivalent Thevenin-based battery model, such as the internal resistance, and therefore these coefficients are different for different batteries.

Significant amount of research work focused on the energy consumption measurement and optimization of different applications and system [22], [23], [24], [26], [28]. A large body of research investigated the energy consumption of mobile devices through profiling, modeling, and debugging [29]. The profiling methods include novel techniques to trace the energy consumption from code to different hardware components [30], [31], [26]. Such profilers also depend on the power consumption modeling. PowerBooster [32] relies on SOC updates and OCV discharge curves to build the regression based power models. V-edge [33] and BattTracker [12] rely on SOC updates, OCV and the load voltage to model power consumption. Sesame [30], AppScope [34], and DevScope [35] rely on SOC updates and current drawn estimates from the Coulomb counter-based fuel gauges. Our approach also depends on battery manager updates. We specifically use SOC update time to find the charging rates and use the battery voltage to determine the length of the CC phase. However, we estimate the FCC of the battery. Our approach works irrespective of the SOC estimation or fuel gauge chip used by the device.

Compared to these related work, we focus on estimating capacity proactively and identifying charging anomaly. Our technique does not require any additional hardware or system modification or battery properties information. Therefore, the approach can be easily implemented as a part of the mobile system and can be integrated with different applications or operating system initiated optimizations.



## IX. FUTURE RESEARCH

The applications of FCC estimation with SOC updates can be quite diverse. For example, the SDB proposed by Badam et al. [17] can include a FCC aware battery scheduling, where SDB will learn the FCC of an individual battery while charging and then schedule the batteries accordingly while discharging. However, our particular research interests are the followings.

1) *SOC Estimation Algorithms*: The proposed FCC estimation technique works regardless of underlying SOC estimation technique. Therefore, implementing and evaluating a new SOC estimation algorithm for cheap and simple fuel gauge chips to reduce SOC fluctuation is one of our future research problems. A simple voltage-based fuel gauge can estimate FCC from the SOC updates while charging and apply it for estimating SOC instead of charging cycles. Similarly a simple coulomb counter-based fuel gauge that requires a complete charging or discharging session for calibrating FCC can instead estimate the FCC even when the battery is partially charged.

2) *Battery Properties Modeling*: We have demonstrated that battery voltage, charging rate, and CC phase vary as the FCC decreases. However, there are no formal models to express such relations yet. In addition, previous studies have shown the relation between FCC, charging current, and charging cycles. Therefore, our plan is to explore and verify various relations further with the crowd-sourced data and additional measurements.

## X. CONCLUSIONS

In this article, we have demonstrated that battery voltage and charging rate together can capture the FCC of Lithium-Ion batteries. Based on this observation, we proposed and validated an online mechanism to estimate recent FCC or FCC loss of a smartphone battery. We also implemented and validated a crowd-sourced mechanism. We found 20-60% of devices of popular models having capacity loss in a large data set of mobile devices. Compared to the traditional approaches, our approach is device based and can be used to debug the performance of smartphone batteries. Finally, this article proposes an algorithm that estimates FCC, charging performance, and detects battery changes. In addition, this work enables modeling and implementing FCC-aware energy optimizations of mobile systems and applications.

## ACKNOWLEDGMENT

This work was funded by the Academy of Finland CUBIC project with grant numbers 277498 and 278207.

## REFERENCES

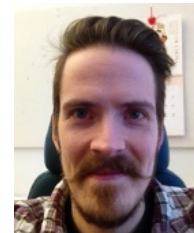
- [1] Nexus help forum. <https://productforums.google.com/forum/#!topic/nexus/D7er7nMbzHI>.
- [2] iPhone 5 battery drops to 20% from around 40% suddenly. <https://discussions.apple.com/thread/5445722?tstart=0>.
- [3] Huge sudden battery drops. <http://androidforums.com/threads/huge-sudden-battery-drops.562600/>.
- [4] Simo Hosio, Denzil Ferreira, Jorge Goncalves, Niels van Berkel, Chu Luo, Muzamil Ahmed, Huber Flores, and Vassilis Kostakos. Monetary assessment of battery life on smartphones. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1869–1880, New York, NY, USA, 2016. ACM.
- [5] Here's How Much a Samsung Galaxy S6 Replacement Battery Costs. <http://www.pcmag.com/article2/0,2817,2481605,00.asp>.
- [6] iPhone 5 Battery Replacement Program. <https://www.apple.com/support/iphone5-battery/>.
- [7] Mohammad A. Hoque and Sasu Tarkoma. Sudden drop in the battery level? understanding smartphone state of charge anomaly. In *Proceedings of the Workshop on Power-Aware Computing and Systems*, HotPower '15, pages 26–30, New York, NY, USA, 2015. ACM.
- [8] Adam J. Oliner, Anand P. Iyer, Ion Stoica, Eemil Lagerspetz, and Sasu Tarkoma. Carat: Collaborative Energy Diagnosis for Mobile Devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, pages 10:1–10:14, New York, NY, USA, 2013. ACM.
- [9] Maxim. Fuel-gauge selection. Technical report.
- [10] Thanh Tu Vo, Weixiang Shen, and A. Kapoor. Experimental comparison of charging algorithms for a lithium-ion battery. In *IPEC, 2012 Conference on Power Energy*, pages 207–212, Dec 2012.
- [11] Yoshinori Kida, Akira Kinoshita, Katsunori Yanagida, Atsuhiko Funahashi, Toshiyuki Nohma, and Ikuo Yonezu. Study on capacity fade factors of lithium secondary batteries using  $\text{LiNi}_{0.7}\text{Co}_{0.3}\text{O}_2$  and graphite-coke hybrid carbon. *Electrochimica Acta*, 47(26):4157 – 4162, 2002.
- [12] Jonghoe Koo, Kitaek Lee, Wonbo Lee, Yongseok Park, and Sunghyun Choi. Batttracker: Enabling energy awareness for smartphone using lithium battery characteristics. INFOCOM '16, New York, NY, USA, 2016. IEEE.
- [13] MAX17048/MAX17049, 1/2-Cell Fuel Gauge with ModelGauge. <https://datasheets.maximintegrated.com/en/ds/MAX17048-MAX17049.pdf>.
- [14] MAX17047/17050, 1-Cell Fuel Gauge with ModelGauge m3. <https://datasheets.maximintegrated.com/en/ds/MAX17047-MAX17050.pdf>.
- [15] Ming Yu and Michael Vega. Impedance track fuel gauge accuracy test for gsm phone applications. Technical report, March 2008. <http://www.ti.com/lit/an/slva455/slva455.pdf>.
- [16] Android Power Profiles. (2016). <https://source.android.com/devices/tech/power.html> Accessed: 2016-11-20.
- [17] Anirudh Badam, Ranveer Chandra, Jon Dutra, Anthony Ferrese, Steve Hodges, Pan Hu, Julia Meinershagen, Thomas Moscibroda, Bodhi Priyantha, and Evangelia Skiani. Software defined batteries. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, pages 215–229, New York, NY, USA, 2015. ACM.
- [18] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. HotCloud'10, Berkeley, CA, USA, 2010. USENIX Association.
- [19] A. Trujillo-Ortiz and R. Hernandez-Walls. skekurtest: Hypotheses test concerning skewness and kurtosis. A MATLAB file. <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3953&objectType=FILE>.
- [20] Deleting outliers. <http://www.mathworks.com/matlabcentral/fileexchange/3961-deleteoutliers>.
- [21] Detection of outliers. <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm>.
- [22] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 280–293, New York, NY, USA, 2009. ACM.
- [23] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, USENIXATC'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.



- [24] Antti P. Miettinen and Jukka K. Nurminen. Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 4–4, Berkeley, CA, USA, 2010. USENIX Association.
- [25] HUSSEIN, A. A.-H., AND BATARSEH, I. A review of charging algorithms for nickel and lithium battery chargers. *IEEE Transactions on Vehicular Technology* 60, 3 (March 2011), 830–838.
- [26] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang. Where is the energy spent inside my app?: Fine grained energy accounting on smartphones with eprof. In *Proceedings of the 7th ACM European Conference on Computer Systems*, EuroSys '12, pages 29–42, New York, NY, USA, 2012. ACM.
- [27] Mohammad A. Hoque and Sasu Tarkoma. Characterizing smartphone power management in the wild. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 1279–1286, New York, NY, USA, 2016. ACM.
- [28] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn A. Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 179–192, New York, NY, USA, 2009. ACM.
- [29] Mohammad Ashraful Hoque, Matti Siekkinen, Kashif Nizam Khan, Yu Xiao, and Sasu Tarkoma. Modeling, profiling, and debugging the energy consumption of mobile devices. *ACM Comput. Surv.*, 48(3):39:1–39:40, December 2015.
- [30] Mian Dong and Lin Zhong. Self-constructive high-rate system energy modeling for battery-powered mobile systems. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 335–348, New York, NY, USA, 2011. ACM.
- [31] Abhinav Pathak, Y. Charlie Hu, Ming Zhang, Paramvir Bahl, and Yi-Min Wang. Fine-grained power modeling for smartphones using system call tracing. In *Proceedings of the Sixth Conference on Computer Systems*, EuroSys '11, pages 153–168, New York, NY, USA, 2011. ACM.
- [32] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 105–114, New York, NY, USA, 2010. ACM.
- [33] Fengyuan Xu, Yunxin Liu, Qun Li, and Yongguang Zhang. V-edge: fast self-constructive power modeling of smartphones based on battery voltage dynamics. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*, pages 43–56, Berkeley, CA, USA, 2013. USENIX Association.
- [34] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulwoo Kang, and Hojung Cha. AppScope: Application energy metering framework for android smartphones using kernel activity monitoring. In *Proceedings of the 2012 USENIX Conference on Annual Technical Conference*, USENIX ATC'12, pages 36–36, Berkeley, CA, USA, 2012. USENIX Association.
- [35] Wonwoo Jung, Chulwoo Kang, Chanmin Yoon, Donwon Kim, and Hojung Cha. Devscope: a nonintrusive and online power analysis tool for smartphone hardware components. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 353–362, New York, NY, USA, 2012. ACM.
- [36] Anthony Barré, Frédéric Suard, Mathias Gérard, and Delphine Riu. A real-time data-driven method for battery health prognostics in electric vehicle use. In *Proceedings of the Second Second European Conference of the Prognostics and Health Management Society 2014*, PHMCE'14, pages 1–8, 2014.
- [37] Shan Yin, Jingyue Pang, Datong Liu, and Yu Peng. Remaining useful life prognostics for lithium-ion battery based on gaussian processing regression combined with the empirical model. In *Proceedings of the Second Second European Conference of the Prognostics and Health Management Society 2013*, PHMCE'13, pages 1–8, 2013.
- [38] Datong Liu, Jingyue Pang, Jianbao Zhou, Yu Peng, and Michael Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectronics Reliability*, 53(6):832 – 839, 2013.
- [39] Min Chen and G. A. Rincon-Mora. Accurate electrical battery model capable of predicting runtime and I-V performance. In *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 504–511, June 2006.
- [40] Zhen Guo, Xinpeng Qiu, Guangdong Hou, Bor Yann Liaw, and Changshui Zhang. State of health estimation for lithium ion batteries based on charging curves. *Journal of Power Sources*, 249: 457–462, 2014.



**Mohammad A. Hoque** obtained his M.Sc degree in Computer Science and Engineering in 2010, and Ph.D in 2013 from Aalto University. He is currently a postdoctoral researcher in Helsinki Institute for Information Technology (HIIT) and University of Helsinki. His research interests include energy efficient mobile computing, data analysis, distributed computing, and resource-aware scheduling.



**Matti Siekkinen** obtained the degree of M.Sc. in computer science from Helsinki University of Technology in 2003 and Ph.D from Eurecom / University of Nice Sophia-Antipolis in 2006. He is currently a postdoctoral research fellow at Aalto University. His current research focuses on mobile computing and networking with a special interest in mobile multimedia services.



**Jonghoe Koo** (S'12) received the B.S. degree in the Department of Electrical Engineering from Seoul National University, Korea in 2011. He is currently working towards a Ph.D. degree in the Department of Electrical and Computer Engineering, Seoul National University, Korea. His current research interests include energy-aware mobile computing and reliable video streaming. He is a student member of IEEE.



**Sasu Tarkoma** (SMIEEE) is a Professor of Computer Science at the University of Helsinki, and Head of the Department of Computer Science. He has authored 4 textbooks and has published over 160 scientific articles. His research interests are Internet technology, distributed systems, data analytics, and mobile and ubiquitous computing. He has seven granted US Patents. His research has received several Best Paper awards and mentions, for example at IEEE PerCom, ACM CCR, and ACM OSR.